GRaNADA: A Network-Aware and Energy-Efficient PaaS Cloud Architecture

Ismael Cuadrado Cordero, Anne-Cécile Orgerie, Christine Morin

GRaNADA: A Network-Aware and Energy-Efficient PaaS Cloud Architecture

Ismael Cuadrado-Cordero
Inria, Rennes, France
ismael.cuadrado-cordero@inria.fr

Anne-Cécile Orgerie
CNRS, IRISA, Rennes, France
anne-cecile.orgerie@irisa.fr

Christine Morin
Inria, Rennes, France
christine.morin@inria.fr

***Abstract*—Current estimated energy usage of data centers and core networks adds up to 3% of the global energy production, while only 42,3% of the population is estimated to be connected. In the last 14 years the number of Internet users has increased tenfold, especially in the period 2010-2014. According to this growing trend, Internet's energy consumption is meant to be a very critical issue in the near future. The emergence of cloud computing represented a major breakthrough in Internet technologies and in reduction of energy consumption. However, due to its centralized nature, this improvement in energy-efficiency has not been reflected in networks' consumption. In this paper, we analyze energy consumption of cloud networks, and present *GRaNADA*, a semi-decentralized Platform-as-a-Service architecture. Through simulations, we show an overall saving much of the energy consumed in standard centralized clouds with our approach.**

I. Introduction

With the emergence of cloud computing [1], the difference between energy consumed and needed in the Internet showed an important decreasing trend. This situation is explained through the use of Virtual Machines (*VMs*) and virtualization methods. They provide adaptable systems, which reduce the need of over-provisioning of active resources, reducing energy consumption in data centers. However, recent energy consumption studies [2] show that this trend discontinued - and even reversed in some cases - in the last years. This situation is partially caused by the heterogeneity of users. A data center located in California might receive requests from users in east Europe, forcing it to keep on enough resources to provide its service - otherwise unutilized during these hours. Similarly, the Internet Service Provider (*ISP*) has to keep a reachable path - which could be shut down - between the clients and this server. This situation has been already addressed by cloud providers like Amazon [3] or Google [4] which changed their conception of cloud computing, splitting their resources in different geographical locations.

This vision of split resources opposes to the typical centralized cloud implementation, where servers are located in the same large data centers. In a centralized approach, according to our measurements, an average French user would need to go through 12 different hops (level-3 network devices), before being connected to the internal cloud network. If the same user is connecting from the USA, it would take only 2 hops to access the same service. Once inside the cloud's network, data are sent from and to different data centers locations according to availability and contextual factors. This is the case of services like Google Drive [5], where two French users working over the same document will have, on average 20 hops between them (10 hops each to the Irish Google's data center). Works like [6] and [7] confirm that, in many cases, information is shared among users located in similar geographical regions. In this case, the use of a centralized system might cause unnecessary delays and packet forwarding outside the network, often referred as "traffic trombone".

On the other hand, fully distributed solutions have been proposed [8]–[10]. However, while decentralized solutions provide great robustness and low latency, they fail to provide simultaneous modification accesses to files. Also, due to replication of content, the use of decentralized cloud systems require a greater bandwidth utilization, as well as additional energy expenses. In the example of Google Drive, the two users would be modifying their own copies of the same file, facing merging conflicts in case of concurrent utilization. Also, in order to keep synchronization of data, a vast flow of information should be continuously exchanged between clients. If the number of participants accessing the document is too large, the required bandwidth might imply the utilization of several paths. Having all those paths on might make the P2P approach less energy-efficient [11] than the centralized one.

The future of cloud computing relies on a better geographical distribution of resources for improving performance and energy-efficiency. Towards this end, we propose *GRaNADA*, a semi-decentralized Platform-as-a-service (PaaS) architecture for real-time multiple-users applications. Our architecture distributes geographically the computation between the clients of the cloud. Thus, energy can be saved by shutting down - e.g. [12] - or downgrading - e.g. dynamic interfaces [13] - unutilized resources such as routers and switches, servers, etc. This solution also provides a lower delay for the user. Along with *GRaNADA*, we propose *DEEPACC*, a cloud-aware routing protocol which distributes the communication between nodes in the network. Our system *GRaNADA* targets services where the geographical distribution of clients working on the same data is limited - for example, a shared on-line document - or those services where, even if the geographical distribution of clients is high, the upload data communication to the cloud is small - for instance a light social network like Twitter. In evaluation, we compare our approach by simulation with 2 existing solutions: replication of data in the edge and centralized private cloud data center architecture. We show that our solution is able to save, in the best cases, up to 75% of the energy in the network.

The remainder of this paper is structured as follows. Section II discusses the suitability of cloud computing in energy-efficiency, existing approaches and the motivation of the present work. In Section III, we introduce our approach, the architecture and the resource management. In Section IV we analyze experimentation results. Finally, Section V highlights our key findings and draws conclusions and directions to future work.

II. Context and Motivation

The exponential increment in the cloud adoption by users has resulted in a growth on providers' infrastructures - and an over-provision of energy and resources. However, even considering the energy inefficiency caused by this situation, the excess of energy spent by providers' over-provision is yet much lower than the excessive energy expense attributed to the use of private data centers. This section builds a energy-efficiency model in cloud computing in subsection II-A and analyses the existent related work and our motivation in subsection II-B.

A. Energy Consumption of Networks in Cloud Computing

Energy-efficiency of data centers has increased since the emergence of cloud computing. However, energy-efficiency of cloud networks - all the networking devices which are used to communicate with and within the cloud - despite being a non-neglectable part of cloud consumption - as it was shown in works like [14] or [15] - has been neglected in bibliography.

The *Total Energy (TE)* statically consumed by a network has been partially or totally modelled in literature [16], [17]. However, in the current work, we model the *TE* strictly under cloud networks' conditions - taking into account the dynamism of cloud systems. In any network, the *TE* is equivalent to the sum of the energy consumed by different paths - *Path Energy (PE)*. *PE* is calculated as the sum of the consumption of all devices needed to communicate between two devices. According to literature, and confirmed by our own experimentation, this value is distributed in *Base Energy (BE)* (energy needed to keep a device on, excluding any kind of performance) and *Configuration Energy (CE)* (values dependent on the variables' configuration needs). The relevant configuration parameters influencing the *CE* have been obtained through our experimentations on different devices. Those are described in Equations 1 and 2. The rest of parameters - number of packets along the interface, IP protocol, etc. - are considered neglectable. One has to note that, according to our measurements and in accordance to literature, traffic itself has a negligeable impact on the energy consumption of network devices such as routers and switches if they do not adapt their configuration according to it.

$$\begin{aligned} Energy_{device} = Base\ Energy_{device} + \\ Configuration\ Energy_{device} \end{aligned} \tag{1}$$

$$\begin{aligned} Configuration\ Energy_{device} = \\ \mathbf{F}\Big(speed\ rate, state, Time, cable\ length, number\ of\ linecards\Big) \end{aligned} \tag{2}$$

The *TE* of a cloud computing network is defined as follows:

$$\begin{aligned} Total\ Energy\ Consumed\ by\ a\ cloud = \\ Energy\ Consumed\ Internally \\ +\ Energy\ Consumed\ Externally \end{aligned} \tag{3}$$

In Equation 3, *Energy Consumed Internally (ECI)* represents the sum of all the energy needed to communicate between devices in the system - either inside the same data center or in different ones. *Energy Consumed Externally (ECE)* represents the energy consumed by the users in order to access the cloud (path along networks external to the cloud provider). Figure 1 shows the internal and external connections in a typical cloud structure.

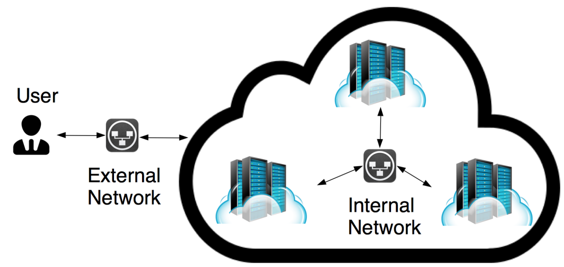


Fig. 1: Scheme of typical cloud interconnection

B. Related work

As the traffic does not directly affect the energy consumption of a network - but indirectly through the CE - the most promising approaches are the re-routing of traffic and the reduction of the length of the path. Thus, in order to significantly reduce *TE*, both *ECI* and *ECE* need to be targetted. *ECI* is assessed by bibliography reducing the distance between *VMs* inside the cloud data center. This way, every unutilized port/device in the network can be switched off. For instance, in [18], authors propose a decentralized networking protocol for cloud data centers. In their work, authors dynamically redirect nodes' communication according to energy measurements.

On the other hand, the reduction of *ECE* entails the reduction of the distance between the data center and the user - that is, physically placing data centers near the user. Following this approach, main cloud providers address this problem by constructing data centers in locations nearby main user's population groups. This is the case of, for instance, Amazon, which owes data centers nearby the most populated areas in several countries, called edged locations. Due to those data centers, VMs can be placed closer to the end users, reducing the length of the path. Data centers locations are decided according to several factors, being 1 or 2 countries enough to supply a whole continent. Also, in emergent markets - like the African continent - it would imply a very specific expansion. Here, the networking problem becomes a geographical distribution of clients' problem.

One of the most appealing geographical based solutions for networks access was proposed by Valancius et al [19]. In this work, authors propose dynamic caching of the most accessed information in ISPs' data centers (called *nano data centers*). Authors show a reduction of overall energy consumption and a significant increase of the Quality of Service (QoS). A comparison between the use of *nano data centers* and centralized data centers is shown in [8]. However, it only translates the problem with several data centers in a much smaller scale. The reduction showed by the authors is around 4% - we will demonstrate that it can be outperformed - and it suffers of a significant replication. Furthermore, the existence of different ISPs providing access to the Internet adds another issue to this approach: communication between ISPs. Finally, this solution lacks of the robustness of distributed approaches, moving the dependency from the cloud data center resources to the ISPs data centers resources, which might be not as well equipped as the cloud providers to face issues like peaks of users.

On the other hand, P2P distributed cloud systems are one of the most accepted solutions in geographical distribution of content. In [20] authors proposed a P2P-based cloud architecture to provide a fully distributed system. Their proposal offers self-organization and management, adaptability and fault tolerance. In their architecture, every client becomes a chunk server, storing part of the data. This solution, however, is not far from typical P2P systems and suffers from issues like replication of data - and the energy consumption associated to those - and delay in data lookup. These issues are found in fully distributed systems.

Also, in [9], authors propose a modification of a P2P-based architecture to run parallel computing. In their solution, authors use a P2P platform specification to construct a reliable, fast and powerful cloud platform, which independently run chunks of code. They define three different roles: user, starting the computation and receiving the final data; central peer, the subnetwork managing the computation (in this case Metadata of MapReduce and backups); and side peer, the computational subnetwork. This solution does not suffer from replication, but it does not consider strong interaction between the clients, therefore excluding multiple-users applications.

P2P-based architectures follow a *Content Delivery Network (CDN)* paradigm. This design casts aside applications based on multiple contributors over the same file - Wikipedia, Google Drive, etc. - which are the scope of the present work. Also, in both described solutions, energy-efficiency is not considered a significant contribution.

III. Our approach

Our approach is based on the closest path policy: bringing the cloud host as close as possible to the client. With this goal, we propose GRaNADA - *GReen Network Aware clouD Architecture* - a semi-decentralized architecture which redistributes computation between participant nodes (including clients and network devices when possible). This way, some nodes obtain the role of providers (which host part of the information), while some others will be regular clients (connecting to the provider to access this information when needed). Along with our architecture, we define a dynamic green routing protocol, which connects the nodes using the less consuming path. To do so, we propose DEEPACC - *Dynamic and Energy Efficient Protocol Adapted for Cloud Computing*. In Subsection III-A, the proposed architecture is presented. In Subsection III-B, a preliminary version of the proposed routing protocol is introduced.

A. GRaNADA

In order to provide an elastic infrastructure aware of the geographical distribution of users, our solution distributes the computational load among all the clients, moving the computation away from the data centers. In *GRaNADA*, a set of clients will directly communicate between them. However, only one of the clients acts as a host for the data and the other clients interact with it. This way, replication is not needed and the problems associated with totally distributed architectures - such as conflicts, network flooding, etc. - are avoided.

We define the concept of *microcloud*, a fully autonomous energy-efficient subnetwork of clients of the same service, designed to keep the greenest path between them. A *microcloud* can be seen as an autonomous set of clients, among which a *Light Virtual Machine (LVM)* is deployed on one of them. The LVM is a partial version of a VM containing only the data needed by the clients in the *microcloud*. It is accessed by the clients belonging to the same *microcloud*.

In order to introduce the concept of *microcloud*, it is necessary to walk through the implications of Equations 3 and 1. Those are:

- There is a direct relation between the number of devices or hops in a network and its energy consumption. In consequence, *microclouds* are distributed in *layers*. A *layer* is a set of clients of the same service, who share the same number of hops to reach the cloud's data center. Two clients might share the same *layer*, if they have the same amount of hops between them and the cloud data center, but belong to different *microclouds*. *Microclouds* can communicate *horizontally* if they belong to the same layer, or *vertically* if they belong to different layers.
- In a much smaller scale, but noticeable, it exists a relation between the traffic and the energy consumed, in the form of configuration profiles. This implies that, given the choice of different configurations inside the same *microcloud*, the greenest one will be taken. Speed of transmission, distance between client nodes or state of the network interface are examples of configuration variables.

The proposed architecture distinguishes four different roles between the members of the *microcloud* plus an additional fifth role, reserved for the cloud data center. The proposed roles are:

- **Client:** This role is inherited by all the members in the *microcloud*, except from the cloud's data center. A client accesses and/or modifies the information following the client-server pattern.
- **Manager:** It controls the access by new clients and the security of the *microcloud*, and acts as a tunnel with other layers. In the case of an unmanageable number of clients, it may start a *microcloud division*, which splits the computation between 2 different *microclouds*.
- **Provider:** It runs the *LVM*, which contains all the information accessed by the clients. Even when only one node acts as a provider, the rest are sorted as backups by the manager. Those with a higher value - at least two, for better redundancy - keep an updated snapshot of the provider's *LVM*. That way, in the event

of a failure, the manager will select one to take its place, thus making the system more robust.

- **Repeater:** Due to the distribution of the network, some client nodes might be closer between them than to the provider. When a client needs to use a different client as a bridge towards the provider, it is called a subscriber to another client. A client which multicasts information from and to subscribers is called a repeater.
- **Cloud data center:** It is installed in *layer* 0, and starts every instance of a *microcloud*. It also serves as a backup for the LVM, when a *microcloud* ends its lifetime.

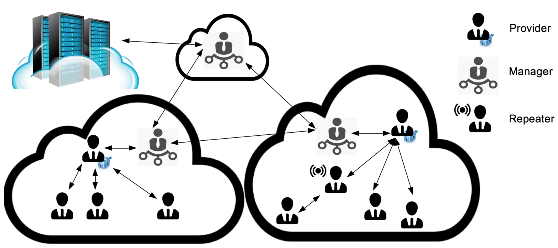


Fig. 2: Scheme of microclouds interconnection

In Figure 2, a scheme of *microclouds* interconnection is shown. The vertical communication of *microclouds* between managers is used as a tunnel to communicate with data centers.

B. DEEPACC

While GRaNADA defines the *microcloud* architecture, the *DEEPACC* routing protocol is used to compute the greenest route between every client node and the provider, and is used as a complement to the resource management during the process of addition and deletion of nodes in the *microcloud*. This protocol is designed to keep the energy consumption inside a *microcloud* to its minimum. To do so, the network is formed following the *greener path* principle. That is, establishing only those connections which consume the least energy. Initially, the clients send the manager - which address obtains using *Dynamic Host Configuration Protocol (DHCP)* - information about their direct connections - known to them using, for example *Border Gateway Protocol (BGP)*.

C. Resource management

The resource management is controlled by the manager of *microcloud*. Every event such as initial formation of *microcloud*, addition or deletion of clients, service failure from the provider or *microcloud divisions* are started and managed by the manager. To do so, it applies different algorithms to dynamically distribute available resources.

When a *microcloud* is started, the algorithm described below starts by a set of nodes: all the clients to be connected and their intermediate network devices. Every node will be considered as a *microcloud* of only one node. The algorithm merges every *microcloud* with the best fit into bigger ones by applying an A* algorithm with heuristic, until the best configuration is reached.

Algorithm 1 Microcloud initial formation: merge function.

Put all nodes in $MicrocloudsList$ (as singleton microclouds).
If $size(MicrocloudsList)$ = 1 **Then**
  Return the first element of $MicrocloudsList$
Remove the first element of $MicrocloudsList$: $mc1$
Remove a random element of $MicrocloudsList$: $mc2$
ForEach possible connection between $mc1$ and $mc2$ **Do**
  Estimate the energy consumption of this connection
  If EC_new connection ¡ EC_best connection **Then**
    best connection = new connection
Create mc_merge the microcloud made of $mc1$, $mc2$ and the least consuming connection
Put mc_merge at the beginning of $MicrocloudsList$
Return merge($MicrocloudsList$)

The location of *LVMs* is decided according to the total energy consumption. This energy is calculated in Equation 4. This value is used in the routing protocol defined in III-B

$$\begin{aligned} &Energy\ Associated\ to\ Provider_A = \\ &(\alpha * \sum_i Configuration_{A,i} + \beta * \sum_i Delay_{A,i}) \\ &*Hardware\ Capabilities_A \end{aligned} \tag{4}$$

where:

- **Configuration** represents the energy consumed by the device when the minimum configuration possible, while managing the load, for every interface;
- **Delay** counts the energy consumed along the necessary time for a packet to its destination;
- and **Hardware capabilities** is a numeric value which represents the capacity of a node to host the LVM, such as spare memory and CPU.

D. Energy efficiency estimation

In this section, the theoretical energy estimation of a *microcloud* is presented. The previous definition of *layer* (i.e. the minimum number of hops between a client and a server) can be applied in the *microclouds* context using the distance from every client to the provider inside a *microcloud*. The internal and external energy consumptions of a *microcloud* can be analyzed as:

$$Energy_{microcloud} = \sum_{l \in Layers} Energy_l \tag{5}$$

where:

$$Energy_l = \sum_{i \in Devices} (Configuration_i\ + Base\ Energy_i) \tag{6}$$

From Equation 5, it can be implied that the sum of the energy consumption of the path between every edge client -

not connected to any other - and the provider constitutes the overall consumption of a network. That is, the sum of the consumption of every *layer* inside the *microcloud*.

$$Configuration_{microcloud} = \int_0^t \sum Traffic\ Energy(t)\ dt \tag{7}$$

As shown in Equation 7, an equivalence can be established between energy consumption and traffic through a network, which is addressed as *Traffic Energy*. This variable is the minimum energy configuration needed to transmit all the messages (including headers, ACKs, etc.) between two nodes. Thus, the configuration energy of a network can be obtained through the addition of the traffic through it along time. Assuming packets splitting in intermediate hops but not merging, we have the number of packets along a network which is always the same or greater than those leaving the edge nodes.

$$Traffic\ Energy \geqslant \int_0^t \Big(\sum_{l \in Layers} Traffic_{Received,l}(t) \Big)\ dt \tag{8}$$

$$\begin{aligned} Traffic\ Energy = \int_0^t \sum_1^N \Big(& Traffic_{Received,i}(t) \\ + \| Traffic_{Received,i}(t) - Traffic_{Sent,i-1}(t) \| & \Big) dt \end{aligned} \tag{9}$$

Equation 8 defines a lower bound of traffic along a network. It is defined as the sum in time of the energy consumed by every packet - energy consumed by every device to receive the packet from the neighbor - which is received by any edge node client. A network should be able to support along a path, at least, the defined traffic flow. If the maximum size of packet along a path is known, it is possible to predict the number of packets that a node will receive from the number of packets sent. That is, knowing the traffic leaving the server, it is possible to predict the traffic received in the clients, provided enough information about the clients, as shown in Equation 9.

Once the traffic flow along every path in the *microcloud* is determined, the configuration energy (*CE*) can be calculated for every node. However, as a traffic redistribution approach, our solution relies on the assumption that ISPs will downgrade as much as possible the performance of the network once freed of the traffic - that is, shutting down unused ports, reducing the broadband according to the traffic needs, shutting down duplicated routers, etc. Our work is designed to reduce the total energy (TE), keeping all the traffic of a service in the same *microcloud*. We achieve energy efficiency by reducing the length of the path (clients connect to the closest provider). This way, we reduce *ECE* by reducing the distance between clients and servers, and we eliminate inefficient internal communication inside the data center (reduce *ECI*) by keeping a centralized management, which reduces this communication - as replication, conflicts solving, etc. - to a minimum. The only internal communication the data center performs is either for the creation or the destruction of a *microcloud*. As for the ISP, the cloud data center does not need to keep as many resources running, because the computation will relay on the *microclouds* distributed among the clients.

IV. Experimentation

The current evaluation compares both the energy utilization and delay in communications of the proposed *microclouds* approach with the main existent solutions. For the experimental evaluation of our *microcloud* approach, we designed a simulation of the French research backbone network *Renater* - for it is representative of a research public network, and its topology is accessible [21] - using NS3 simulator [22] to mimic the network, and ECOFEN [23] to evaluate the energy consumption. Over the *Renater* backbone, we added 2 extra client nodes per interface, to enhance the network. The repeaters were modelled as client which multicast information between the provider and one or more clients. In total, it sums up to 142 different nodes, with a homogeneous distribution of energy consumption.

We compare three different approaches in our experiments: the current cloud architecture, the greenest and most utilized approach in bibliography and ours, in terms of both energy consumption and delay between clients and server. We acknowledge the existence of other approaches not contemplated in the evaluation. However, they either escape the energy efficiency scope of this paper or they are already represented by any of the solutions. An example is found in load distribution among datacenters, which does not differ from the centralized simulation shown on the first protocol.

- **All ON:** This approach assumes every network device and link is being utilized, and keeps them all in an active state. In this protocol, the addition of a new client would be instantaneous, but at a really expensive price.
- **SPO (Shortest Path Only):** The implementation of this approach is a fully informed version of OSPF [24]. It starts from a fully shut down network and, for every node, before starting any communication, calculates the shortest path between the sender and the central node of the backbone: Paris in Renater's case. Only nodes in the resulting path are switched on, if they were not yet. It represents the nano data centers approach. It responds well to addition of clients in the working path, but not under routers which are shut down.
- **DEEPACC:** Implementation of the described approach. It starts from a fully shut down network and, for every client node, before starting any communication, calculates the shortest path between the sender and the closest node in the *microcloud*. Only those nodes in the resulting path are switched on, if they were not yet. In current experimentation, it has only being considered a large *microcloud*. Splitting of *microcloud* has been left for future work. The addition of new clients is never immediate.

For the trace used in this simulation, a real 45 minutes trace has been obtained from an actual Google Drive session, using the network packet analyzer tool Wireshark [25]. This trace is used in every client.

A. Clients and data centers modeled energy consumption

1) Client energy consumption: For the client energy consumption, and due to the lack of a suitable survey on clients' energy consumption in literature, we have obtained experimental values through our own model. This model is used to give a rough approximation of the overall energy savings of the proposed solution, and is not meant to be precise. For our experiments, we have used a MacBook Pro - Retina, 15-inch, early 2013 with 8 GB 1600 MHz DDR3 memory and a 2.7 GHz Intel Core i7 4-cored processor. We have estimated the capacity of the battery in Joules, and compared the length in battery life under two different utilization profiles:

- **Average utilization:** Utilization of the computer running average energy consuming applications (browse Internet, music play, etc.);
- **VM simulation:** Similar utilization to the previous one, while running a simulation of GRaNADA over a VM. The VM used runs Debian OS, with 8MB base memory and 16 processors (no graphic interface running).

In average, we observe a difference of utilization between the two experiments of 47.3 Watts (67.79 W under average utilization and 115.10 W with the addition of the VM). This result is consistent with the specifications of maximum energy utilization of the manufacturer (about 200 Watts).

2) Data center energy consumption: On the other hand, works like [26]–[28] set the energy consumption of an average 5,000 sq feet in 1,127 kW per hour (27,048 kW per day) with an average PUE (Power Usage Effectiveness) of around 1.8. This configuration is relevant, for the trend in cloud providers is either to build data centers of this size in different locations or to divide each data center in rooms of about the same size.

B. Core network consumption

We performed two different set of experiments:

1) Comparison of energy consumption of the three different approaches with all users in the same *layer*;
2) Evolution of energy consumption of approaches when the number of users increases. This experiments set has been separated in random and sequential selection of users. In a random selection, users are chosen using a pseudo-aleatory algorithm, while in a sequential selection, all users are selected due to proximity between them; that is, the nodes in the *microcloud* are concentrated in a geographical point.

Figure 3 shows the energy consumption of the 3 protocols, assuming all the clients are 1 hop distant from each other. In the case of *All ON*, all the devices in the network are working and responsive. In the case of *SPO*, only those devices in the working path are working and responsive. The energy saving of this protocol respect to the former one is of almost 90% less energy than *All ON*. Finally, *DEEPACC* is the protocol which behaves better from an efficient point of view: consuming 75% less energy than *SPO*.

Figure 4a shows the evolution in consumption of protocols *SPO* and *DEEPACC*, when the number of participant client nodes is increased. In this experiment, client nodes are randomly chosen among the available client nodes in the network, and all inactive nodes are shutdown. It is appreciated how the consumption of *DEEPACC* is always smaller or equal to *SPO*. That is caused by the nature of both protocols, being the first a reduction of the path between the client node and the *microcloud* (minimum path until any node in the *microcloud*), and the former a reduction of paths between every node and the main router. That is, in the worst scenario, the shortest path between every node in the cloud is the same as the shortest path to the main node - for example, in the case of two nodes split by the main node. In this case, the energy consumed by *DEEPACC* is exactly the same as the one consumed by *SPO*. However, this is a very unlikely case in a backbone network, and even more when the number of nodes is increased. Therefore, the number of working nodes in a *microcloud* will be always smaller or equal to the number of working nodes in a totally centralized system, even in a nano data centers system.

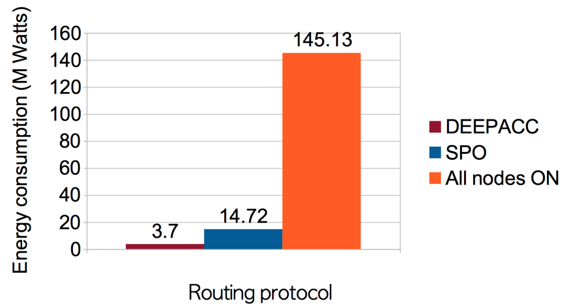
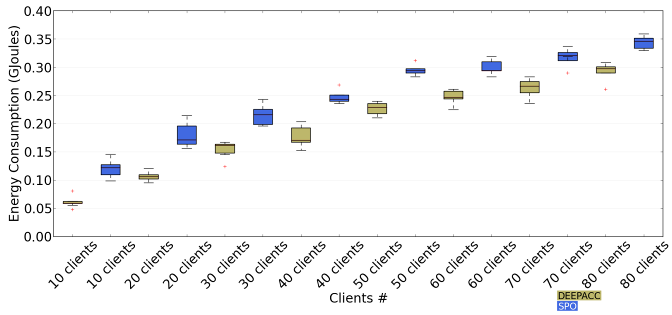


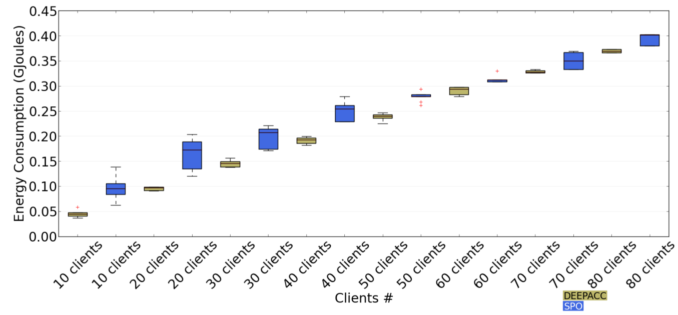
Fig. 3: Energy consumption of 1 *layer* communication under different protocols

Figure 4b shows the same set of experiments run activating only nearby client nodes - according to the minimum number of hops between them. Again, it is shown that *DEEPACC* exhibits always a better energy-efficiency than *SPO*. The result of these experiments is explained because of the same circumstances as the previous ones. Due to the nature of the protocols, energy consumption of *DEEPACC* protocol can only be equal or lower than *SPO*. It is worth noticing that the difference between energy consumptions decreases when the number of client nodes increases. That is because, being the nodes sequential, most of the paths are re-utilized.

Experiments described in Figures 4 and 5 show that the average delay between client nodes and the LVM using *DEEPACC* is also reduced due to the physical proximity of clients. As expected, the delay in communication lines between clients has to be included as part of the routing metric. Also, under a sequential activation of nodes the delay function behaves more predictably than under a random activation.In Figure 5b, the evolution of the average delay per packet of protocols *SPO* and *DEEPACC* under a sequential addition of client nodes is shown. As expected, the delay using *DEEPACC* is smaller. The reason of this behavior is the proximity of all involved nodes. In Figure 5a, the evolution of the delay of protocols *SPO* and *DEEPACC* under a random addition of client nodes is shown. Under *DEEPACC* protocol, every two nodes in the network will connect through the shortest path (independently

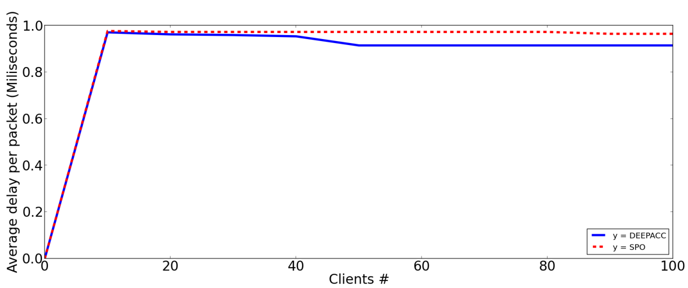


(a) Random increase of active nodes

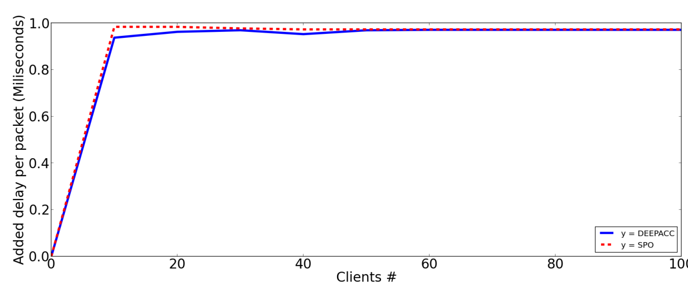


(b) Sequential increase of active client nodes

Fig. 4: Energy comparison between SPO and DEEPACC protocols over a random increase of active client nodes



(a) Random increase of active client nodes



(b) Sequential increase of active client nodes

Fig. 5: Delay comparison between SPO and DEEPACC protocols over a random increase of active client nodes

if the central node is included in the path). It implies that the delay between every two nodes using *DEEPACC* will be always equal or smaller than using *SPO*.

C. Extrapolation

In average, the energy saved using *DEEPACC*, compared to a centralized system is of 42 GWatts only in Renater's backbone. From a strictly energy point of view, this implies that the network can host up to 898,340 clients (6,416 clients per node in the network), all of them hosting the LVM - playing the provider role - before being less energy-efficient than the regular cloud architecture.

Using the figures described above for cloud data centers (1,127 kW), we get that the client side can run the equivalent of 23,978 LVMs with the same energy consumption. It is equivalent to running 8 VMs per server (assuming a central data center of 5,000 sq feet with 100 racks and 30 servers per rack) or creating 23,978 microclouds - more if the LVM is split. Summing up the network savings, we reach 922,167 microclouds (6,587 clients per node, assuming them all play the providers' role) before the energy expenses overcome the energy savings.

D. Discussion

As a semi decentralized approach, the current approach deals with manageability, resource allocation, and security and privacy issues. It is well accepted that a centralized system is better in terms of manageability which, in decentralized systems, can become a problem as the number of users in the network grows. This issue is addressed in GRaNADA by centralizing the management of microclouds. This way, the manageability of a microcloud is centralized and self contained, so it can be seen as a centralized approach.

Secondly, GRaNADA has to face resource allocation issues. As the network grows, it does so the heterogeneity of devices in it. Therefore, not every device might be suitable for hosting the LVM. In the process of selection of the provider, the system takes into account the capabilities of the devices in the network, to choose the most suitable provider. However, to avoid affecting the user's experience, if the infrastructure supports it the service may be hosted in a network device, such as routers [29].

Finally, privacy and security issues should be dealt with. These issues are left to the isolation and measures of the virtualization hypervisor, for every microcloud service is centralized in one provider. Thus, the same privacy and security measures included in centralized cloud systems can be extended into its microcloud version.

V. Conclusions and Future Work

Cloud computing is one of the most used technologies in the last few years, and it represents a great part of the Internet's daily traffic. In this paper, we explore the energy consumption of cloud computing technologies.

We have proposed a new cloud computing architecture GRaNADA, and a specific protocol DEEPACC to reduce energy consumption in clouds' networks. This architecture is a semi-decentralized PaaS and it exhibits interesting properties in terms of QoS and especially latency. Simulations show that, using the GRaNADA, one can save up to 75% of the spent network energy compared to a centralized cloud computing approaches. Our approach is also more energy-efficient than the most popular semi-decentralized solutions, like nano data centers. Values obtained in experimentation are explained by the reduction of the number of hops between users, and simulated over a private network. Even when a public network can be shared by different users accessing different services, it still consumes less energy through the reduction of needed broadband and the consequent possible downgrade of systems.

Results of experimentations are encouraging, but they leave aside some questions. Our system is designed to reach its best in heterogeneous networks - where the routing path and provider's selection are more important -, with a dynamic consumption where the configuration of network devices (such as routers and switches) can be optimized. Next steps in experimentation will be studying a wider range of network infrastructures with different applications, dynamic energy configurations and heterogeneous consumption. Also, we plan to introduce prediction of utilization in order to reduce the overhead time in additions of new users.

Acknowledgments

This work was partially supported by the Inria Project Lab Discovery, an Open-Science Initiative aiming at implementing a fully decentralized IaaS manager: http://beyondtheclouds.github.io.

References

[1] J. Baliga, R. Ayre, K. Hinton, and R. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, Jan 2011.

[2] D. Realty, "N. america campos survey results," "http://content.digitalrealty.com/sfc/servlet.shepherd/version/download/06880000000k573", 2013, survey.

[3] Amazon, "Aws global infrastructure," "http://aws.amazon.com/about-aws/global-infrastructure/", 2015.

[4] Google, "Google cloud," "https://cloud.google.com/", 2015.

[5] ——, "Google drive," "https://www.google.com/drive/", 2015.

[6] D. Hristova, M. Musolesi, and C. Mascolo, "Keep your friends close and your facebook friends closer: A multiplex network approach to the analysis of offline and online social ties," *CoRR*, vol. abs/1403.8034, 2014. [Online]. Available: http://arxiv.org/abs/1403.8034

[7] M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the flickr social network," in *WWW*, 2009.

[8] F. Jalali, R. Ayre, A. Vishwanath, K. Hinton, T. Alpcan, and R. S. Tucker, "Energy consumption of content distribution from nano data centers versus centralized data centers," *SIGMETRICS Performance Evaluation Review*, vol. 42, no. 3, pp. 49–54, 2014.

[9] Z. Peng, H. Ting-lei, L. Cai-xia, and X. Wang, "Research of p2p architecture based on cloud computing," in *Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on*, Oct 2010, pp. 652–655.

[10] A. Lawey, T. El-Gorashi, and J. Elmirghani, "Distributed Energy Efficient Clouds Over Core Networks," *Lightwave Technology, Journal of*, vol. 32, no. 7, pp. 1261–1281, April 2014.

[11] J. Pierson, *Large-scale Distributed Systems and Energy Efficiency: A Holistic View*, ser. Wiley Series on Parallel and Distributed Computing. Wiley, 2000.

[12] Juniper Networks, "SRX High Availability Design Guide," http://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/technotes/SRX%20High%20Availability%20Deployment%20Guide.pdf, 2014.

[13] ——, "Dynamic Interfaces and Dynamic Subscriber Interfaces Overview," http://www.juniper.net/techpubs/en_US/junose15.1/topics/concept/dynamic-subscriber-interfaces-overview.html, 2014.

[14] J. Buysse, K. Georgakilas, A. Tzanakaki, M. D. Leenheer, B. Dhoedt, C. Develder, and P. Demeester, "Calculating the minimum bounds of energy consumption for cloud networks," in *ICCCN*, H. Wang, J. Li, G. N. Rouskas, and X. Zhou, Eds. IEEE, 2011, pp. 1–7.

[15] A.-C. Orgerie, M. D. d. Assuncao, and L. Lefevre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 47:1–47:31, Mar. 2014.

[16] C. Lange, D. Kosiankowski, R. Weidmann, and A. Gladisch, "Energy consumption of telecommunication networks and related improvement options," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 17, no. 2, pp. 285–295, March 2011.

[17] K. Gomez, D. Boru, R. Riggio, T. Rasheed, D. Miorandi, and F. Granelli, "Measurement-based modelling of power consumption at wireless access network gateways," *Computer Networks*, vol. 56, no. 10, pp. 2506 – 2521, 2012, green communication networks.

[18] A.-C. Orgerie, L. Lefèvre, and I. Guérin Lassous, "On the Energy Efficiency of Centralized and Decentralized Management for Reservation-Based Networks," in *Proceedings of the Global Communications Conference, GLOBECOM 2011, 5-9 December 2011, Houston, Texas, USA*, 2011, pp. 1–5.

[19] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the internet with nano data centers," in *ACM Conference on Emerging Networking Experiments and Technology (CoNEXT)*, 2009, pp. 37–48.

[20] K. Xu, M. Song, X. Zhang, and J. Song, "A cloud computing platform based on p2p," in *IT in Medicine Education, 2009. ITIME '09. IEEE International Symposium on*, vol. 1, Aug 2009, pp. 427–432.

[21] "Renater," https://www.renater.fr.

[22] "Ns3 simulator," "https://www.nsnam.org/", 2015.

[23] A.-C. Orgerie, L. Lefevre, I. Guerin-Lassous, and D. M. L. Pacheco, "ECOFEN: An End-to-end Energy Cost mOdel and Simulator For Evaluating Power Consumption in Large-scale Networks," in *Proceedings of the 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, ser. WOWMOM '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1–6.

[24] J. Moy, "OSPF Version 2." [Online]. Available: https://tools.ietf.org/html/rfc2328

[25] "Wireshark," https://www.wireshark.org/.

[26] E. N. Power, "Energy logic: Reducing data center energy consumption by creating savings that cascade across systems," "http://www.emersonnetworkpower.com/documentation/en-us/latest-thinking/edc/documents/white\%20paper/energylogicreducingdatacenterenergyconsumption.pdf", 2008.

[27] ——, "San diego supercomputer center at the university of california," "http://www.emersonnetworkpower.com/documentation/en-US/Brands/Liebert/Documents/Case\%20Studies/CH-00025_final.pdf", 2011, case of study.

[28] P. Jonathan G. Koomey, "Growth in data center electricity use 2005 to 2010," "http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterelectuse2011finalversion.pdf", 2011.

[29] Juniper Networks, "Creating innovative embedded applications in the network with the junos sdk," https://www.juniper.net/us/en/local/pdf/whitepapers/2000378-en.pdf, 2011.